Systolic array-based CNN accelerator for gesture recognition

CSEE 4840 - Embedded Systems - Spring 2025

Hang Ye
Columbia University
New York City, NY
hy2891@columbia.edu

Xuanmin Zheng
Columbia University
New York City, NY
xz3372@columbia.edu

Sicheng Hua
Columbia University
New York City, NY
sh4605@columbia.edu

Yu Jia
Columbia University
New York City, NY
yj2839@columbia.edu

Siyuan Li
Columbia University
New York City, NY
sl5590@columbia.edu

ABSTRACT

This proposal outlines the development of a hardware-accelerated hand gesture recognition system optimized for the DE1-SoC FPGA platform. The system will utilize a resource-efficient Tiny-VGG CNN architecture to classify 8 distinct hand gestures from video frames. By implementing this streamlined neural network directly in hardware, we aim to achieve real-time performance within the strict resource constraints of the DE1-SoC. The proposed accelerator design emphasizes efficient use of DSP blocks, on-chip memory, and logic elements while maintaining acceptable recognition accuracy. This implementation will demonstrate the viability of complex computer vision tasks on resource-limited FPGA platforms.

1 INTRODUCTION

Hardware acceleration of convolutional neural networks (CNNs) enables sophisticated gesture recognition capabilities to be deployed on embedded systems. This project focuses on implementing an efficient Tiny-VGG CNN accelerator on the Terasic DE1-SoC platform to provide real-time hand gesture recognition capabilities.

1.1 Motivation

While software implementations of gesture recognition require significant computational resources, hardware acceleration on FPGAs offers several advantages:

1. Significant reduction in power consumption for embedded applications
2. Real-time processing capabilities through parallel computation
3. Customization of the processing architecture for the specific application
4. Lower latency compared to GPU or CPU implementations
5. Potential for integration into portable or wearable devices

The DE1-SoC platform presents both opportunities and constraints that shape our accelerator design approach, requiring careful optimization to achieve high performance within the available resources

2 SPECIFICATION

2.1 Assessment

The hardware-accelerated gesture recognition system will be evaluated based on hardware resource utilization (logic elements, DSP blocks, and memory usage on DE1-SoC), processing throughput (frames per second and gesture recognition latency), power efficiency (power consumption relative to software implementation), recognition accuracy (classification performance within hardware constraints), and the resource-accuracy tradeoff (optimal balance between FPGA resource usage and recognition performance).

2.2 Milestones

The project will be completed in several phases:

1. Model Architecture Design and Quantization (Week1)
   - Design compact Tiny-VGG CNN architecture for 8-gesture classification
   - Quantize model to fixed-point representation (8/16-bit precision)
   - Simulate and validate quantized model accuracy
2. Hardware Accelerator Design (Week2
   - Design optimized convolution engine utilizing DE1-SoC DSP blocks
   - Create efficient pooling and activation function hardware units
   - Implement weight buffer and feature map memory management
   - Design system controller and interface to ARM HPS

(3) FPGA Implementation and Resource Optimization (Week3-4

- Map design to DE1-SoC resources
- Optimize critical paths to meet timing constraints
- Implement tiling strategies to work within memory limitations

(4) System Integration and Performance Evaluation (Week5

- Integrate accelerator with camera interface and display output
- Measure and analyze resource utilization metrics
- Benchmark processing throughput and power consumption
- Compare with software baseline implementation

(5) Documentation and Final Optimization (Week6

- Create detailed documentation of hardware architecture
- Finalize resource utilization and performance report
- Benchmark processing throughput and power consumption
- Develop user guide for system deployment and usage

2.3 Critical Aspects

DE1-SoC Resource Management The DE1-SoC platform imposes strict constraints that shape our accelerator design:

(1) Logic Element Utilization
Available: Approximately 85K Logic Elements Target usage: <70Logic-efficient implementation of activation functions Resource sharing between similar computational blocks

(2) DSP Block Allocation
Available: 112 DSP blocks (18x18 multipliers) Optimization strategy: Maximize MAC operations per DSP block Time-multiplexing of DSP resources across layers Evaluation of DSP vs. LUT-based implementation tradeoffs

(3) Memory Architecture
On-chip memory: .45Mb total (Block RAMs) External SDRAM: Higher capacity but with access latency Layer-by-layer processing to fit within memory constraints Weight compression techniques to reduce storage requirements Ping-pong buffer implementation to overlap computation and data transfer

(4) Bandwidth Considerations
HPS-FPGA bridge bandwidth limitations Minimization of off-chip memory access Data reuse strategies to reduce memory bandwidth requirements Burst transfers for improved SDRAM efficiency

(5) Tiny-VGG Accelerator Architecture
Convolution Engine Design
Parallel MAC units organized for maximum DSP utilization Line buffer implementation for sliding window operations Dataflow optimization to maximize data reuse Support for different kernel sizes without reconfiguration

(6) Processing Optimizations
Zero-skipping to avoid unnecessary computations Layer fusion to reduce intermediate data storage Resource sharing across convolution layers Pipelining for improved throughput

(7) Quantization Strategy
8-bit fixed-point representation for weights and activations Per-layer scaling factors to maintain accuracy Custom activation function implementation Rounding policy optimized for hardware implementation

(8) System Control and Interfaces
AXI-based interface to ARM processor DMA controllers for efficient data transfer Command queue for processing control Lightweight driver implementation on ARM HPS

REFERENCES

[1]Gunduz, A., Kose, N., Rigoll, G. (2019). Real-time hand gesture detection and classification using convolutional neural networks. Proceedings of the IEEE International Conference on Automatic Face Gesture Recognition (FG), 2019, 1-8.

[2]Lai, K., Yanushkevich, S. N. (2018). CNN+RNN depth and skeleton based dynamic hand gesture recognition. Proceedings of the 24th International Conference on Pattern Recognition (ICPR), 2018, 3458-3463.

[3]H. T. Kung, "Why Systolic Architectures?," IEEE Computer, vol. 15, no. 1, pp. 37–46, 1982.

[4]Terasic Technologies. "DE1-SoC User Manual (rev.F/rev.G Board)," Version 2.0.4, January 28, 2019.